**Cookie Inspector Application**

**Abstract**: The Cookie Inspector Application is designed to help users monitor and manage the security of cookies downloaded by their web browsers. Available either as a Chrome extension or a terminal-based tool (final distribution TBD), the application automatically triggers whenever a new cookie file is downloaded. It parses the cookie data and evaluates its security by checking for known vulnerabilities, such as insecure attributes, excessive expiration periods, etc. Based on the computed security score, the user is prompted to delete cookies that fall below a safe threshold. For cookies deemed secure, no interruption occurs; however, a log of all scanned cookies is maintained for user reference and convenience. This application provides a proactive approach to managing web privacy and security with minimal user effort.

Plan is to get a working CLI prototype in Python because I can hit the ground running and figure out if this is an easy or hard idea to implement. From there I can determine if I should learn JavaScript/TypeScript to get the added perk of deploying this idea to the Google Chrome Extension Store.

**Tech Stack:**

* Python
* PathLib
* SQLite

**List of System Requirements**:

* **Parse Chrome cookie database** - Read cookie files from user's cookie cache
* **Security score calculation** - Assign security score to each cookie based on attributes
* **Flag insecure cookies** - Identify files with well-known vulnerabilities and prompt user for action if required.
* **Command-line interface** – let the user decide whether to let the program persist or scan specified file.
* **Scan reporting** - Display summary of total cookies scanned, issues found, actions taken
* **Basic logging** - Keep record of scan dates and major actions for user reference

Goals for next meeting:

* Get program architecture diagram ready
* Start writing a python script

**Project Structure:**

**Core**:

cookie\_listener.py

main.py (activate listener or scan all current cookies or scan individual file (like a search))

create\_cookie\_score.py

**Features**:

cookie\_general\_descriptive\_stats.py + query.sql (makes the stats possible)

logPastCookieScans.json

cookie\_test.html

chrome\_cookie\_extractor.py (returns all downloaded cookies for system scan)

To-do

1. Would like to make a .env file that creates a virtual python environment before each runtime to make a less error prone application.
2. Every time I run the script it asks for system passcode so find a way to make that less annoying for the user like a once and done kind of deal.
3. Write test classes
4. Two separate scores for meta data and one for when I crack encryption

// quarantine aspect: do you want the program to quarantine

Update:

* Listener class is complete
* Start working on the control flow
* Finish encryptions (AES)

A screenshot of a computer

AI-generated content may be incorrect.